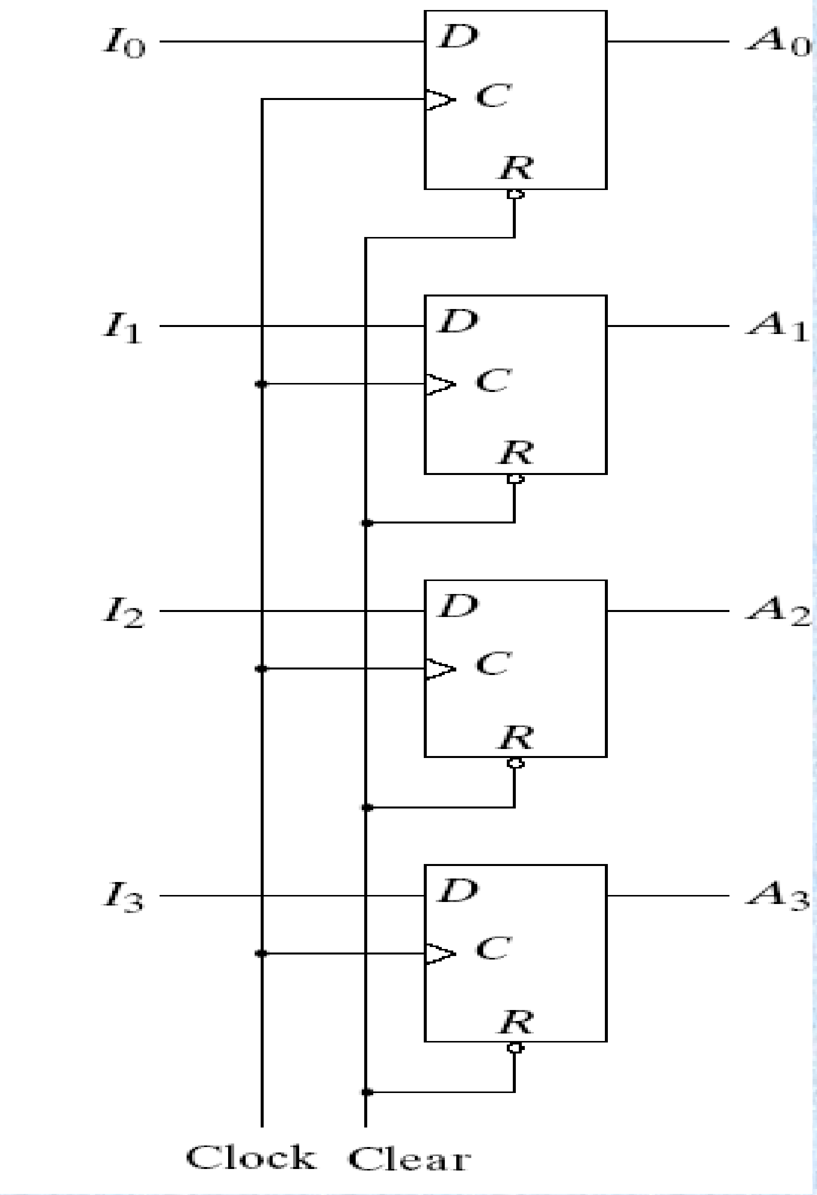
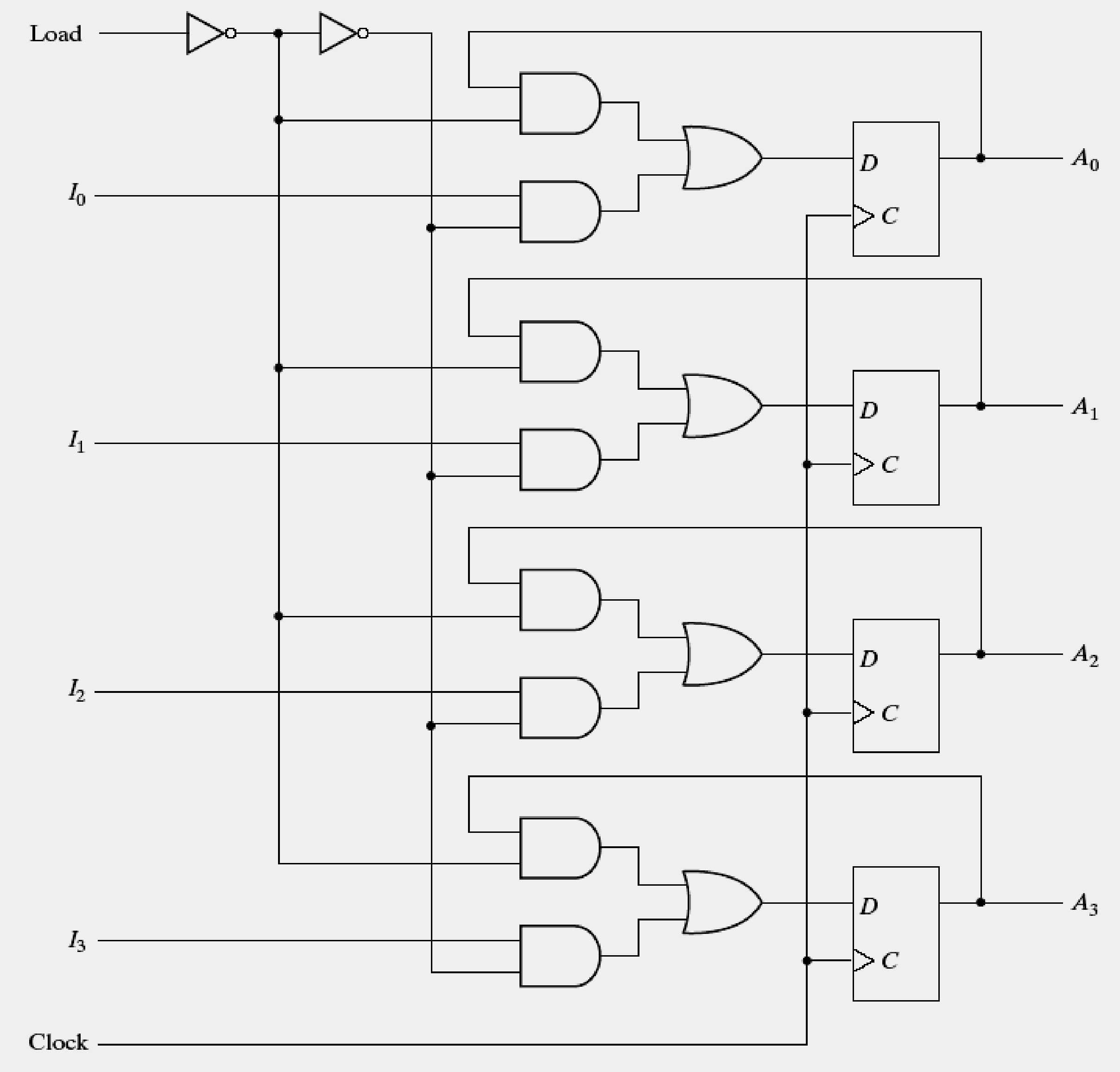
Chapter 4

Registers and Counters

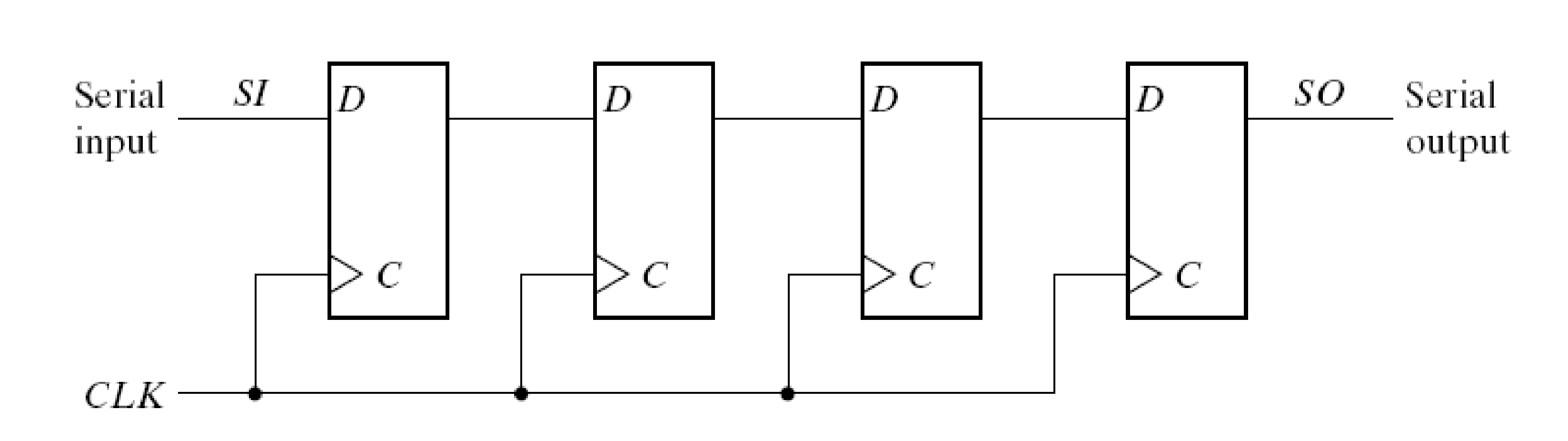
Registers



Register with Parallel Load



Shift Right Register



SI	Reg. contents	SO
1	→ 1 0 0 1	
1	→ 1 1 0 0	(1)
1	→ 1 1 1 0	(0)
1	→ 1 1 1 1	(0)

Oral:

1010 and that the SI is 0 all the time.

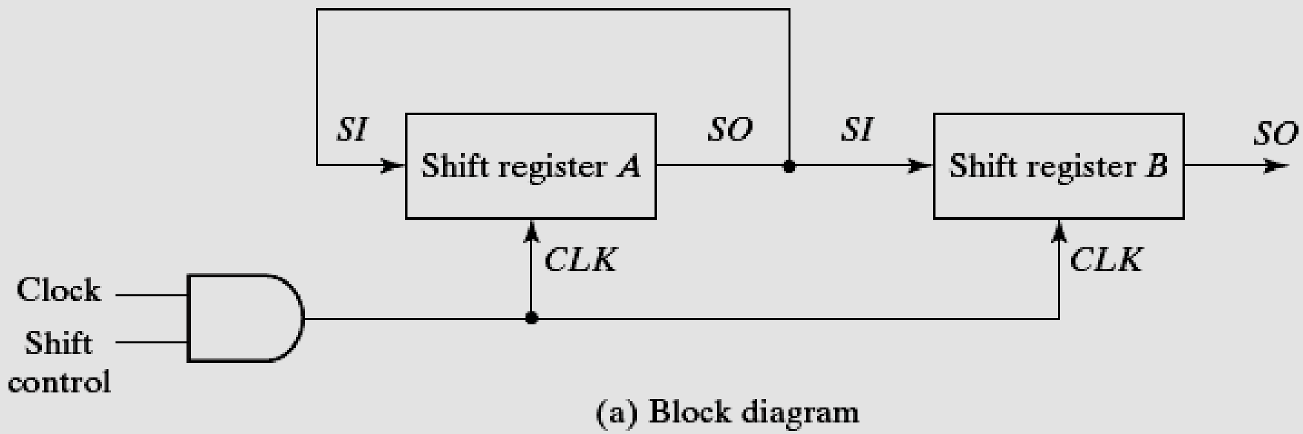
Shift Left Register

- **Try to draw a 4-bit shift right register.**

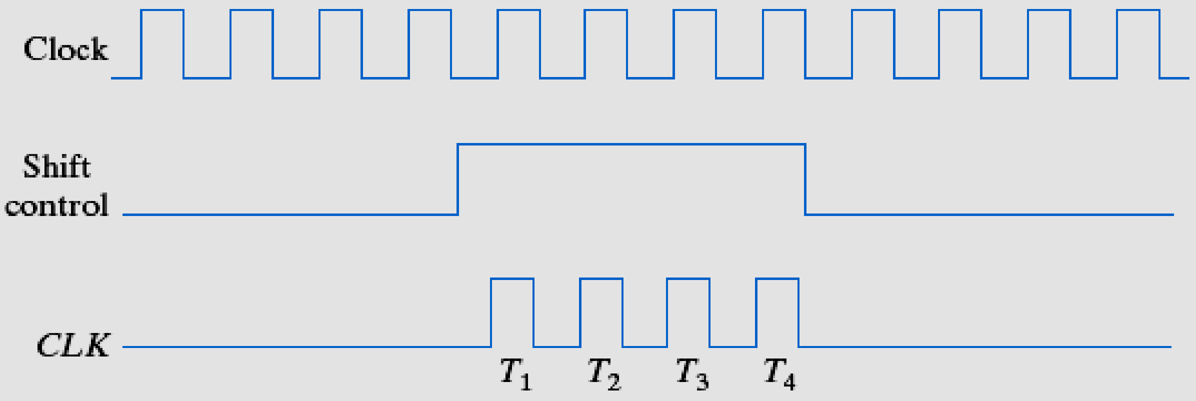
So	Reg. contents				SI
	1	1	0	1	1
(1)	1	0	1	1	1
(1)	0	1	1	1	1
(0)	1	1	1	1	1

- **Now try to do the same if you know that the current value in the register is 1010 and that the SI is 0 all the time.**

Serial Transfer



(a) Block diagram



(b) Timing diagram

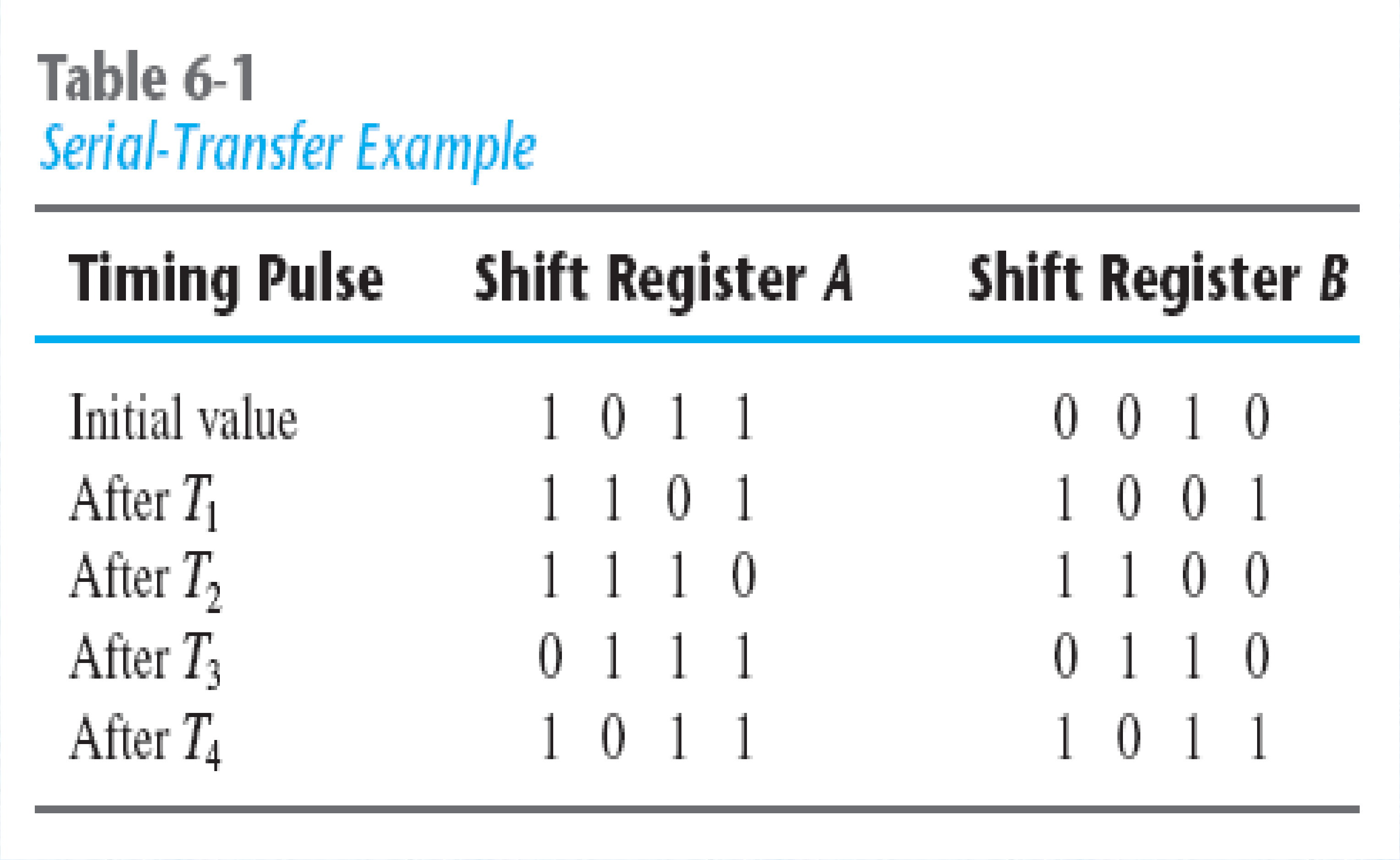
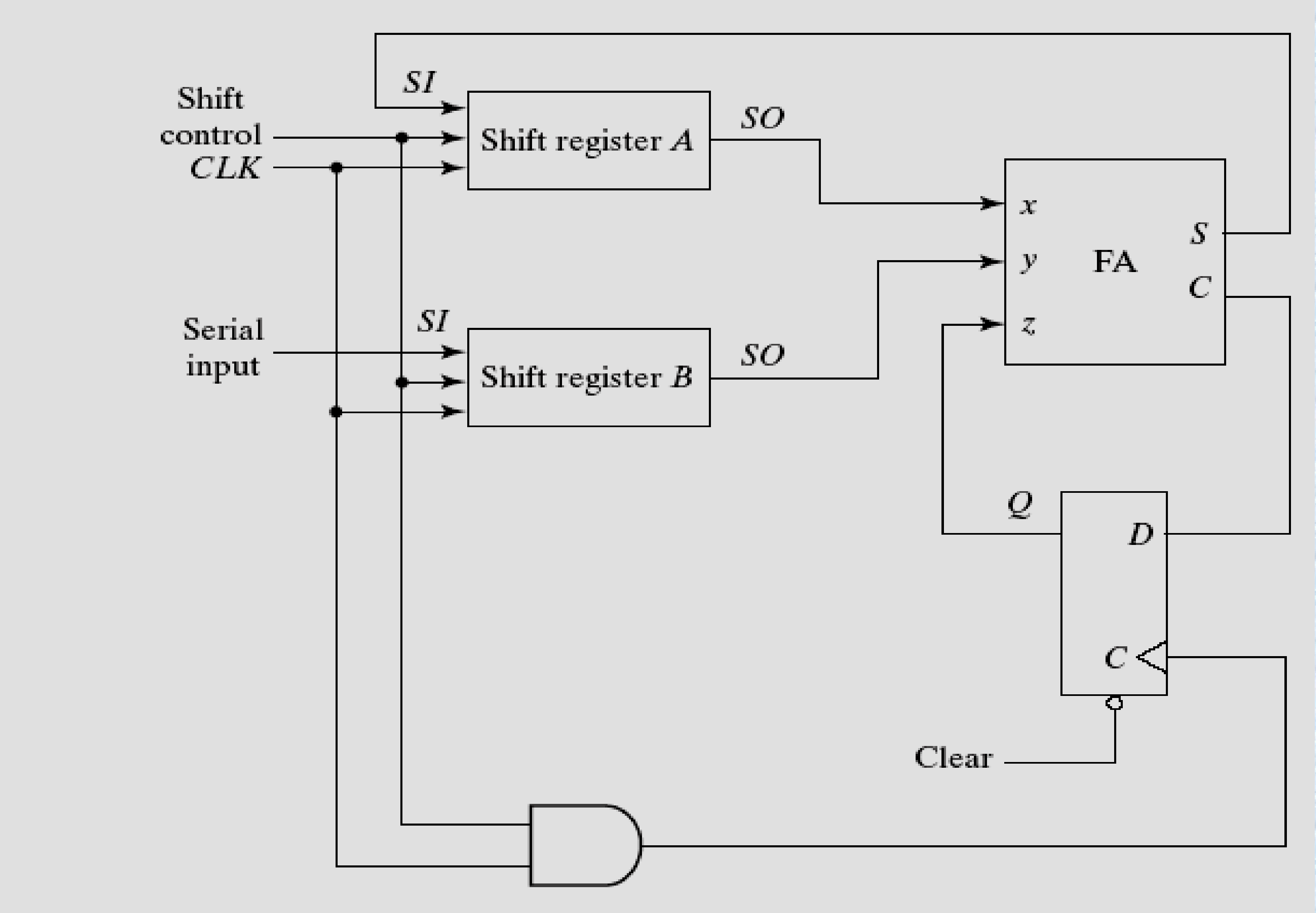


Table 6-1
Serial-Transfer Example

Timing Pulse	Shift Register *A*	Shift Register *B*
Initial value	1 0 1 1	0 0 1 0
After T_1	1 1 0 1	1 0 0 1
After T_2	1 1 1 0	1 1 0 0
After T_3	0 1 1 1	0 1 1 0
After T_4	1 0 1 1	1 0 1 1

Draw a parallel transfer circuit from register A to register B each having 4-bits

Serial Addition



Universal Shift Register

The most general shift register has the following capabilities:

1. A *clear* control to clear the register to 0.
2. A *clock* input to synchronize the operations.
3. A *shift-right* control to enable the shift right operation and the *serial input* and *output* lines associated with the shift right.
4. A *shift-left* control to enable the shift left operation and the *serial input* and *output* lines associated with the shift left.
5. A *parallel-load* control to enable a parallel transfer and the *n* input lines associated with the parallel transfer.
6. *n* parallel output lines

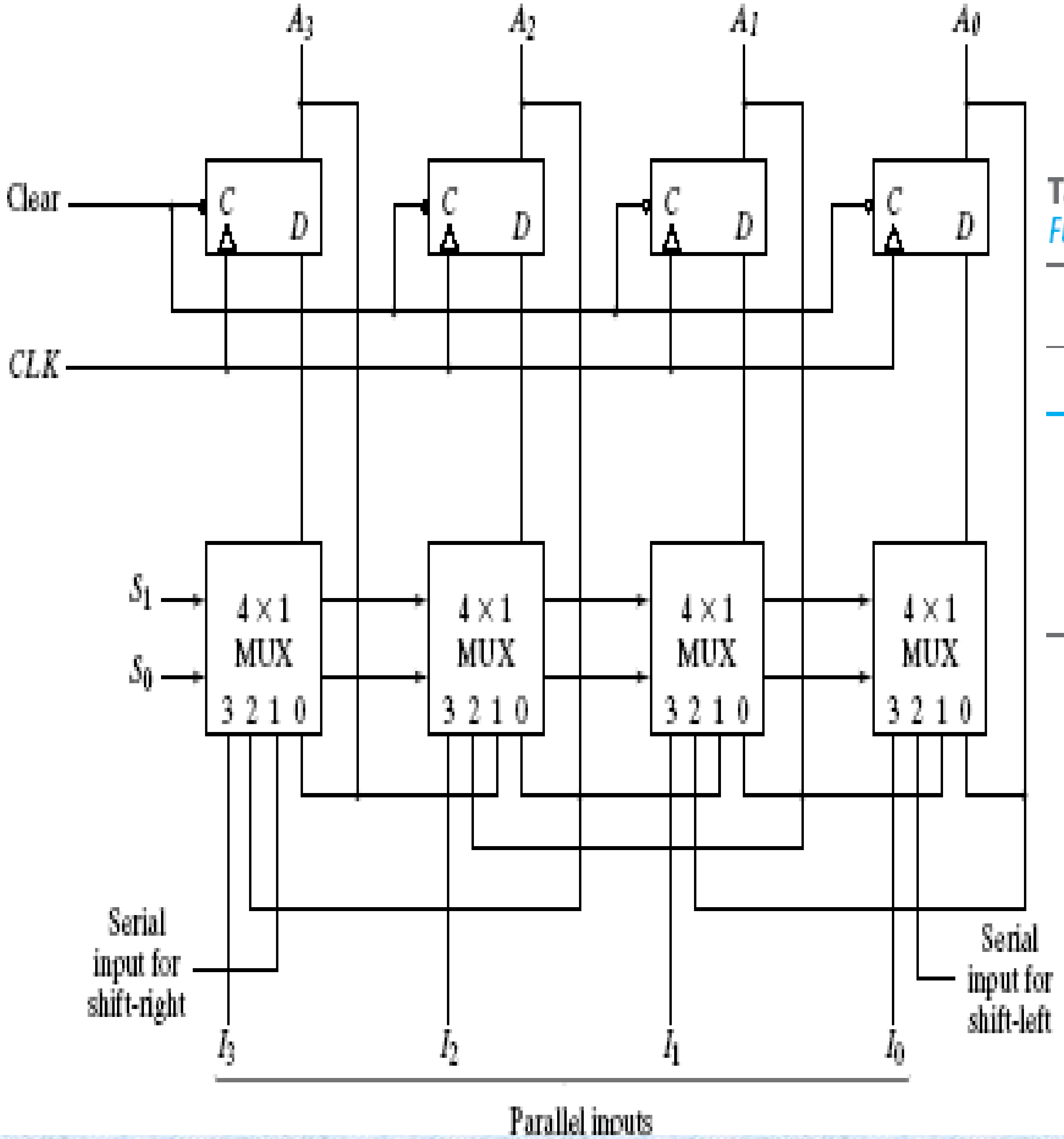
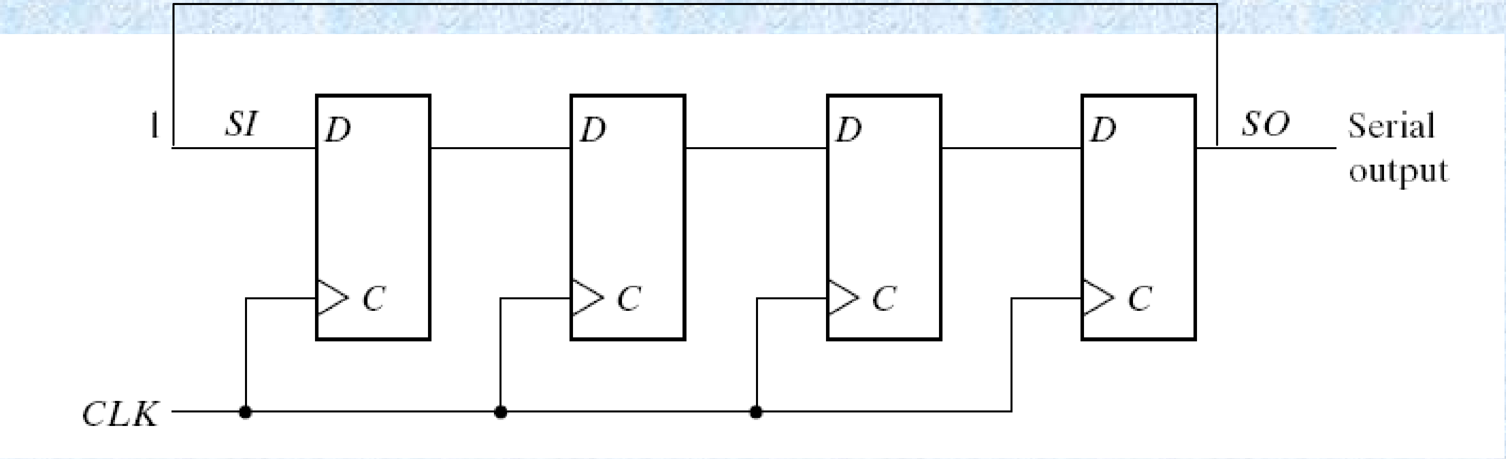


Table 6-3
Function Table for the Register of Fig. (

Mode Control		
s_1	s_0	***Registe***
0	0	No chan
0	1	Shift righ
1	0	Shift left
1	1	Parallel l

Shift-Circulate Register



Draw 4-bit Circular Shift left Register

Counters

- A counter is essentially a register that goes through a predetermined sequence of states.
- The gates in the counter are connected in such a way as to produce the prescribed sequence of binary states.
- Although counters are a special type of register, it is common to differentiate them by giving them a different name.
- Counters are available in two categories: ripple (Asynchronous) counters and synchronous counters.

Asynchronous (ripple) and synchronous counters

- In a (asynchronous) ripple counter, the flip-flop output transition serves as a source for triggering other flip- flops.
- The *C* input of some or all flip-flops are triggered not by the common clock pulses, but rather by the transition that occurs in other flip-flop outputs.
- They use different clock pulse for each flip flops.

- In a synchronous counter, the *C* inputs of all flip-flops receive the common clock.
- Synchronous counters are more popular.

Binary Ripple Asynchronous Counters

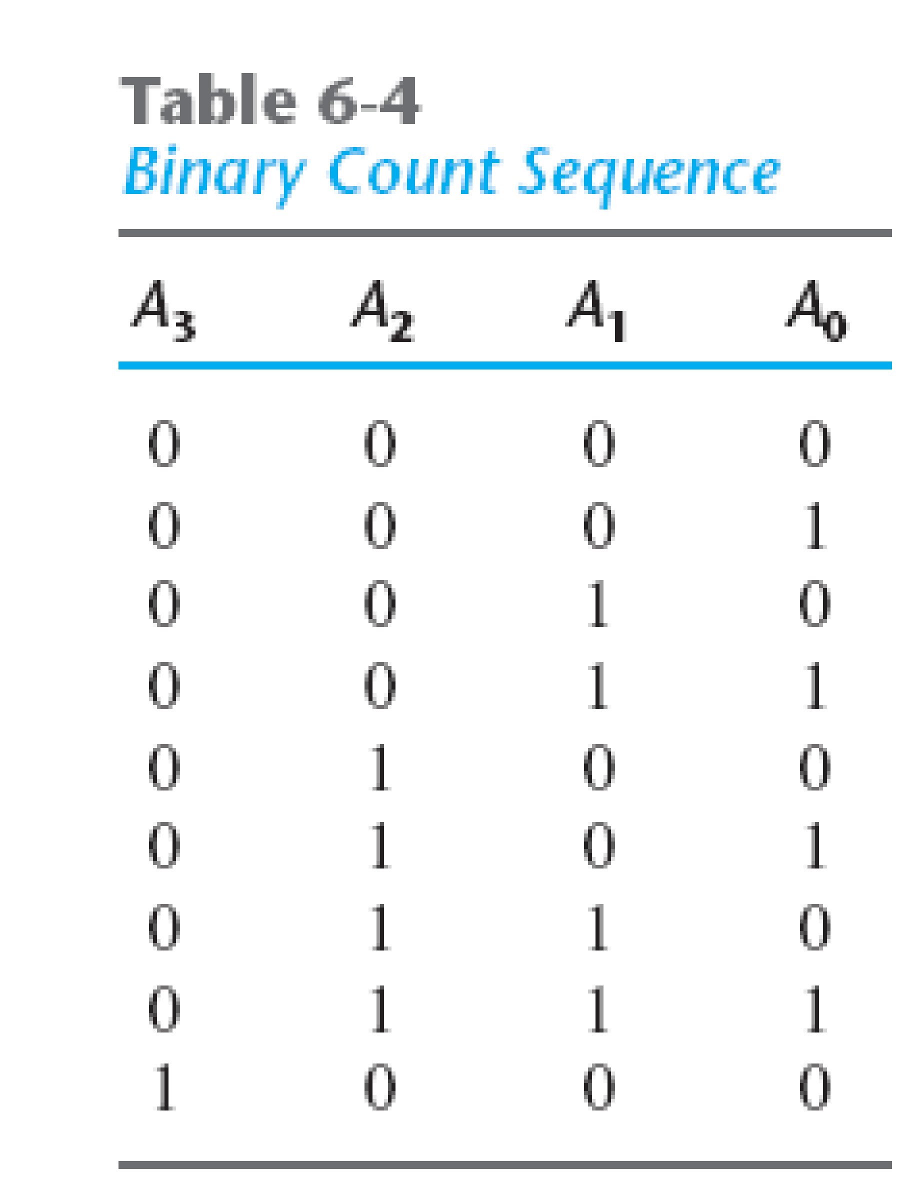
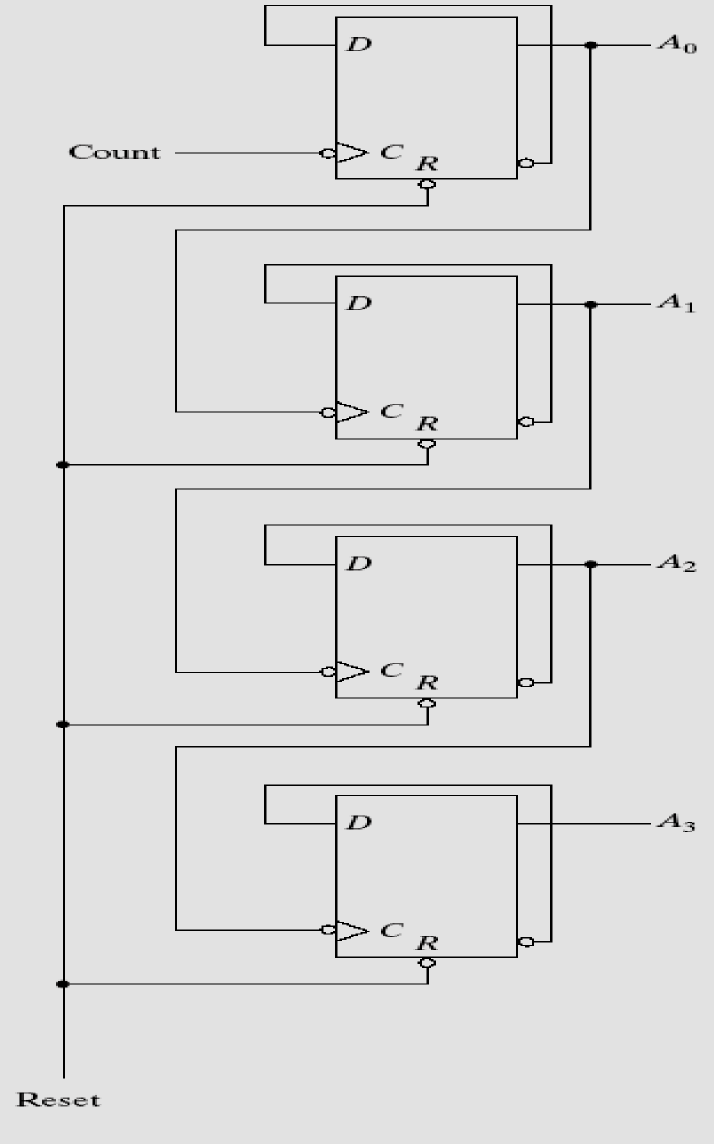
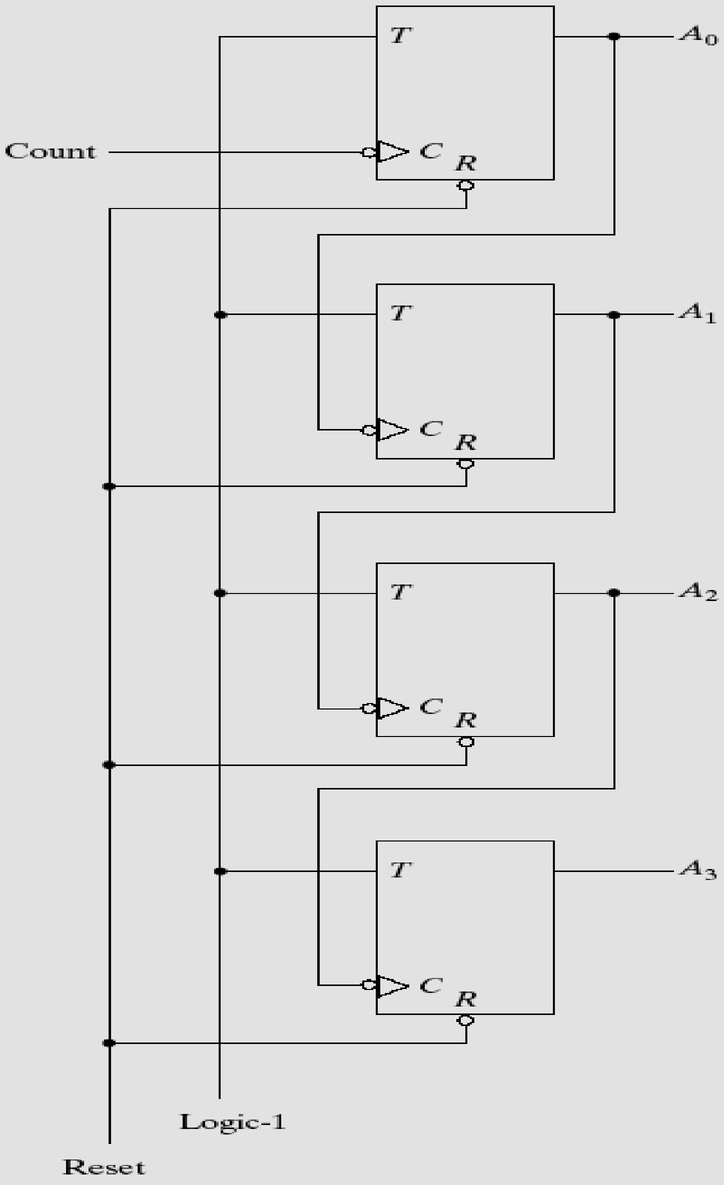


Table 6-4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0



(b) With D flip-flops



(a) With T flip-flops

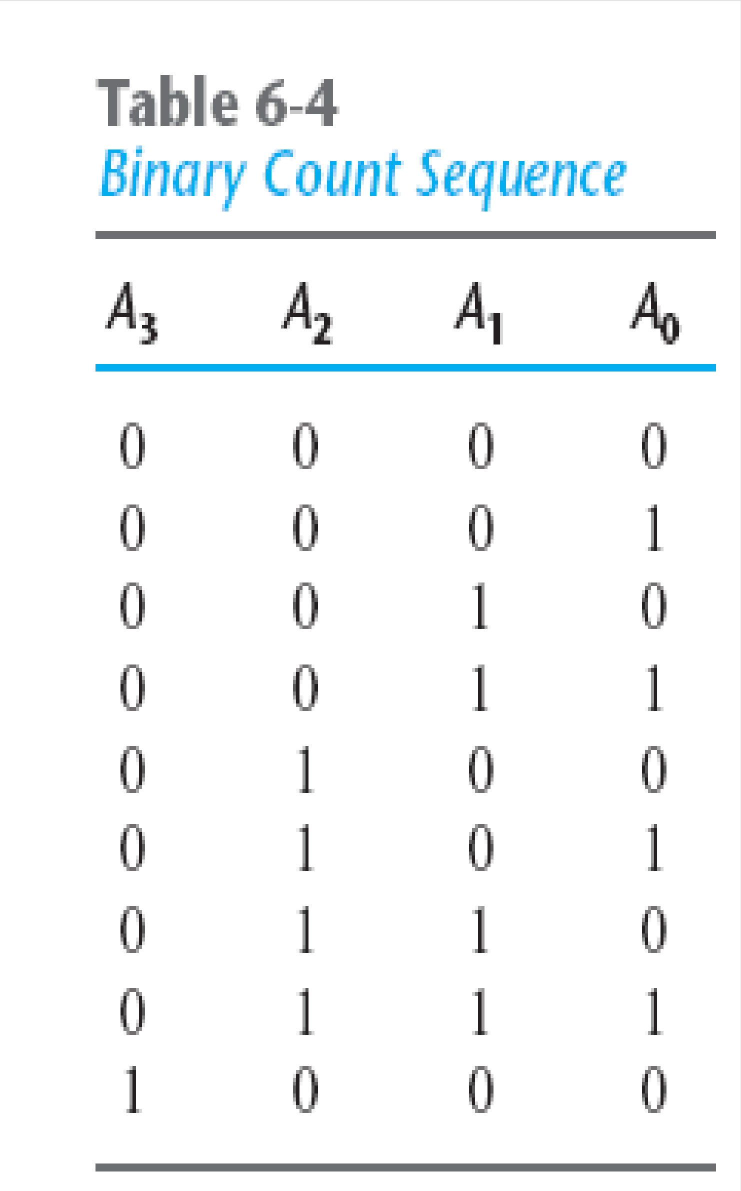


Table 6-4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

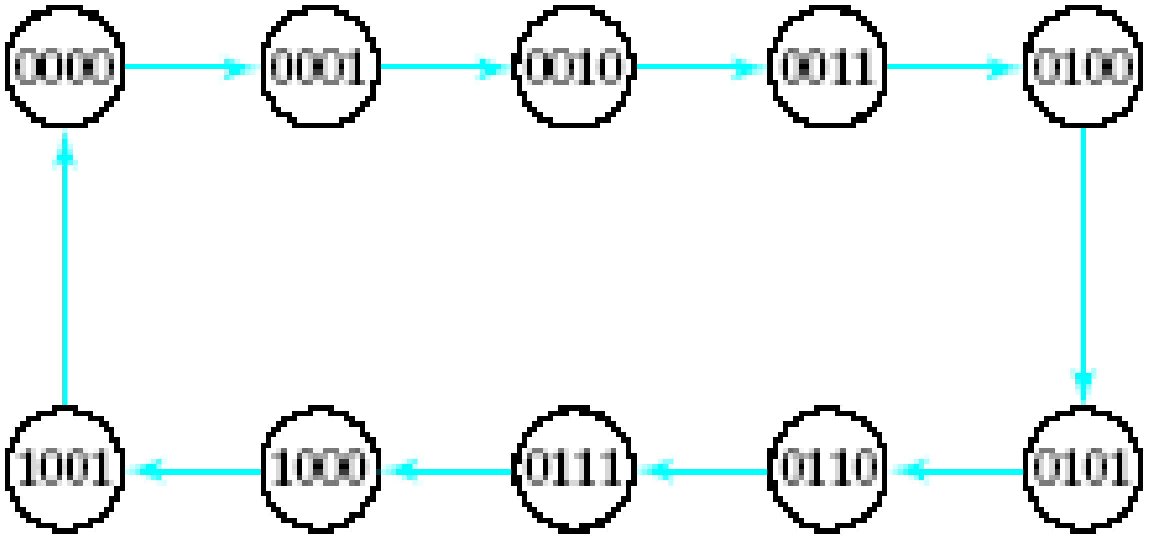
(b) With D flip-flops

Binary count down counter

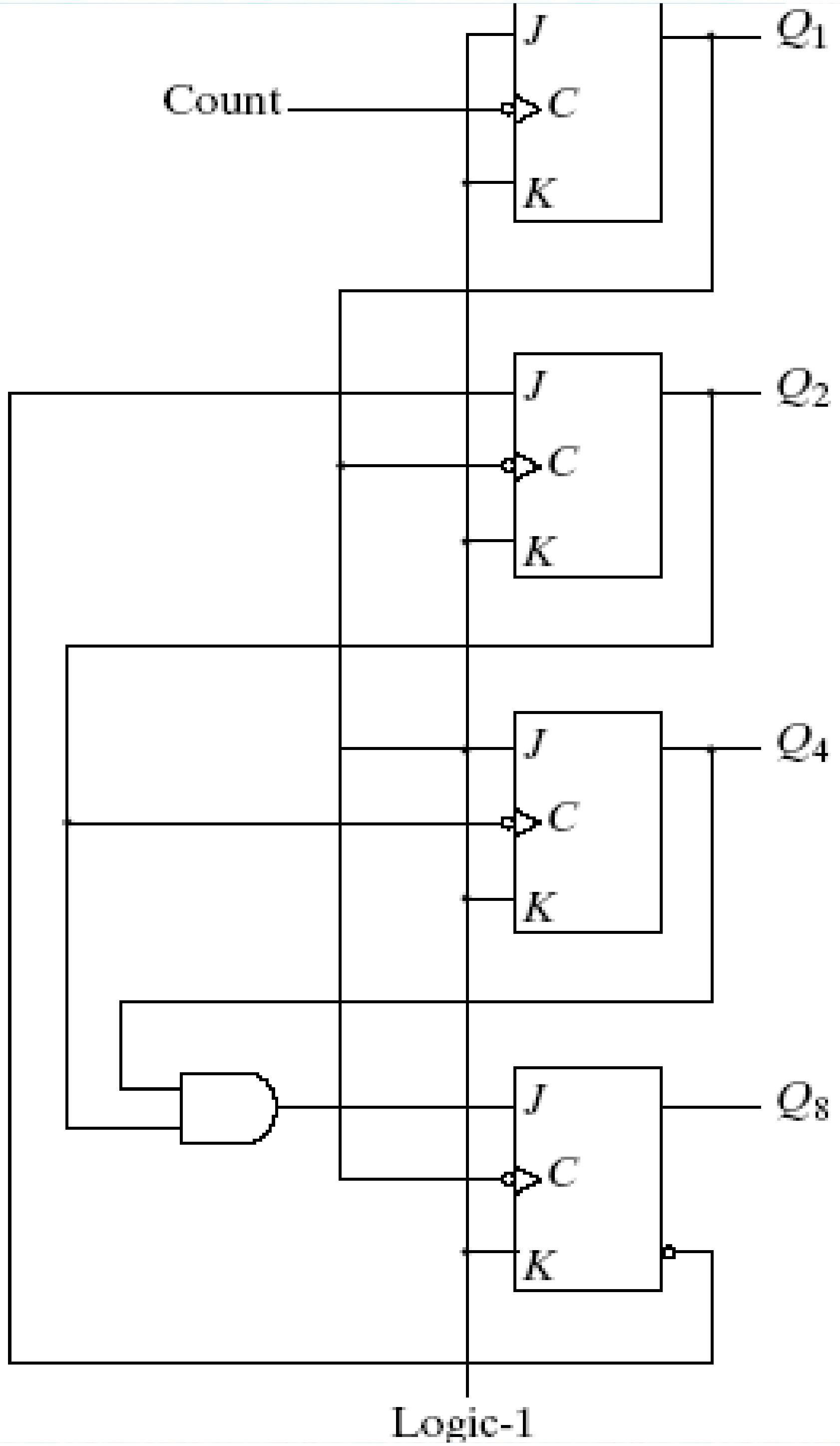
- the diagram of a binary count-down counter looks the same as in previous Fig., provided all flip-flops trigger on the positive edge of the clock.
- The circle in the *C* inputs must be absent.
- If negative-edge-triggered flip-flops are used, then the *C* input of each flip-flop must be connected to the complement output of the previous flip-flop.

- Draw it

BCD Ripple Counter



Q8	Q4	Q2	Q1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1



- *Q*1 changes state after each clock pulse (after count signal comes).

- *Q*2 complements every time *Q*1 goes from 1 to 0 as long as Q8=0 (from 0000 to 0111).

- When *Q*8 becomes 1 (at 1000), *Q*2 remains at 0.

**Q*4 complements every time *Q*2 goes from 1 to 0.

**Q*8 remains at 0 as long as *Q*2 or *Q*4 is 0.

*When both *Q*2 and *Q*4 become 1 (at 1001), *Q*8 complements when *Q*1 goes from 1 to 0.

* *Q*8 is cleared on the next transition of *Q*1.

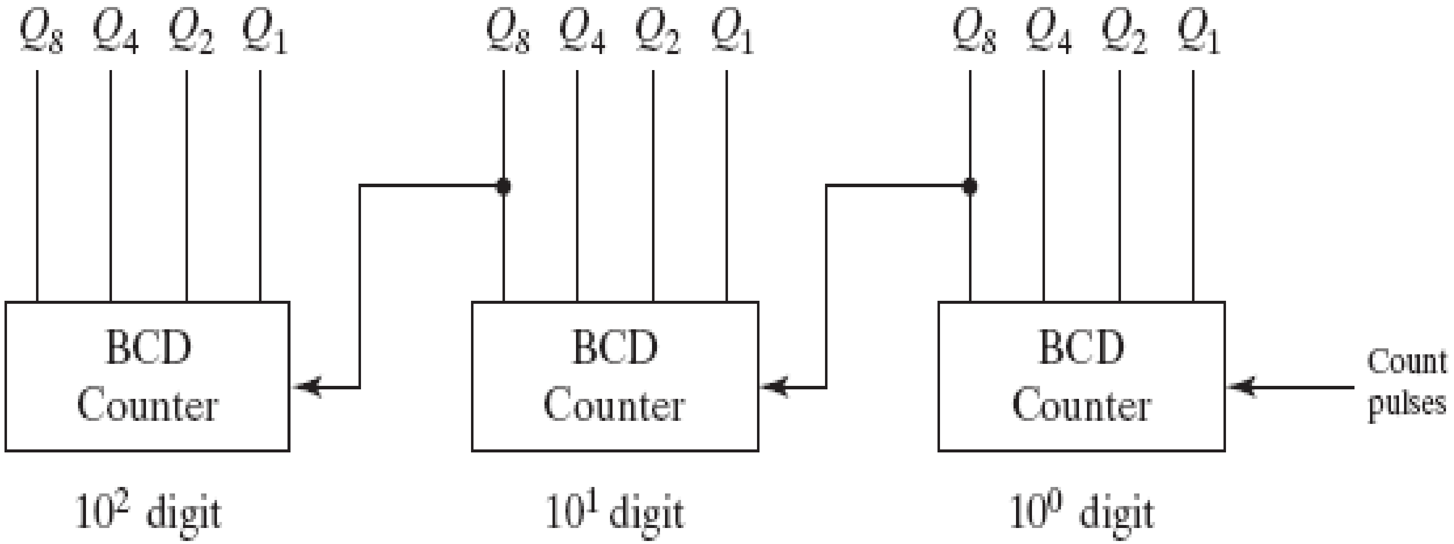
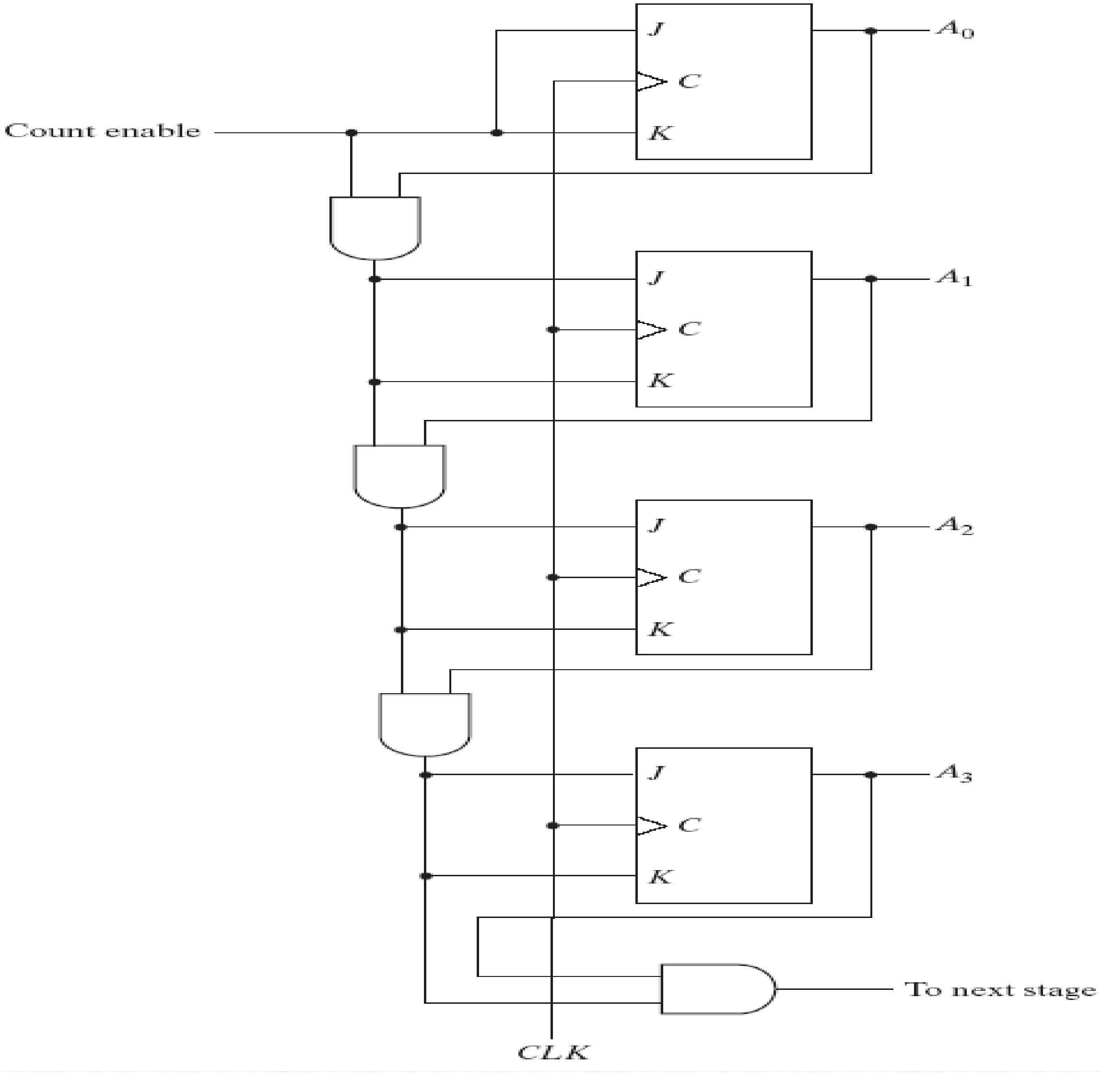


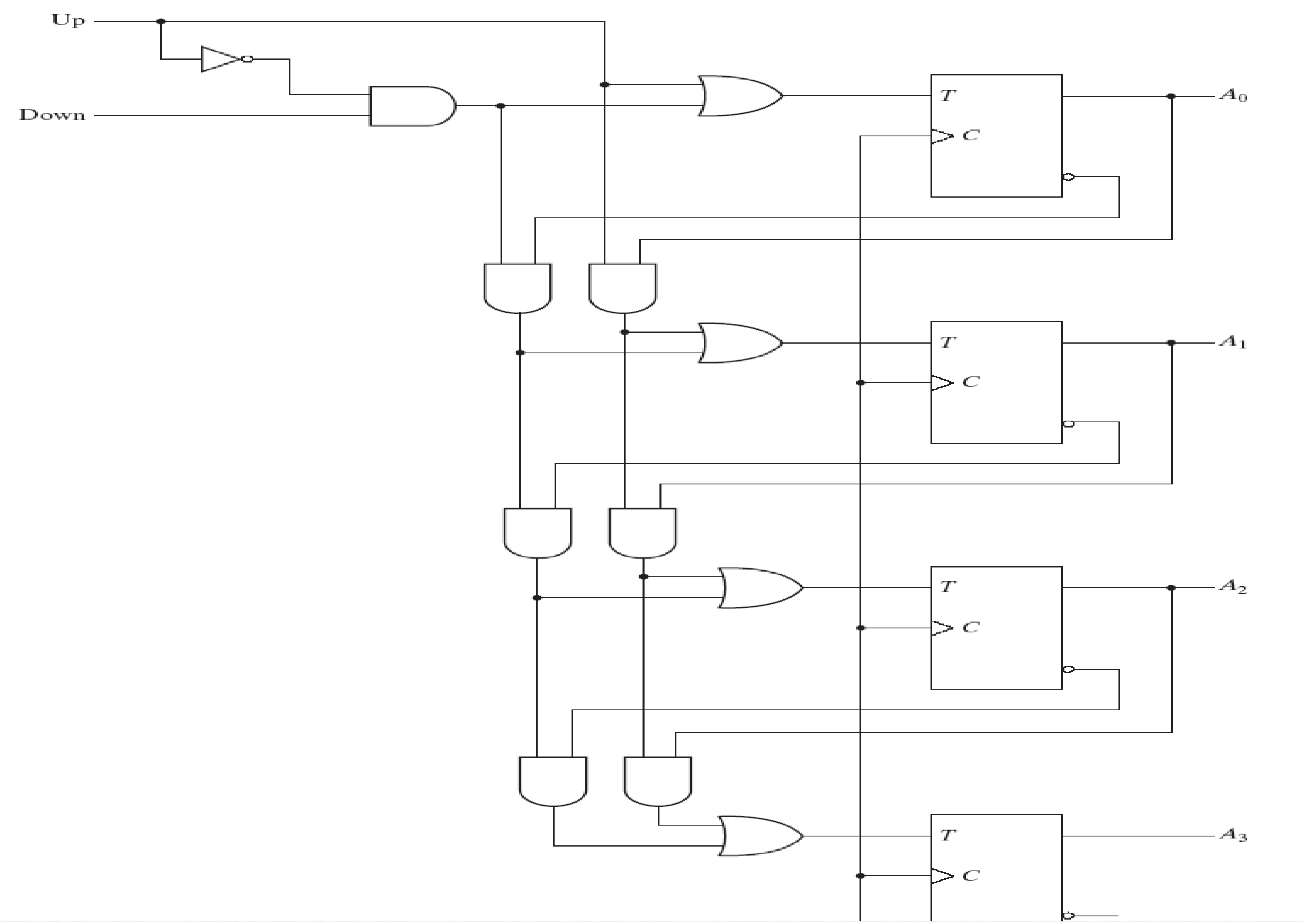
FIGURE 6-11

Block Diagram of a Three-Decade Decimal BCD Counter

Binary Synchronous Counter



Up-Down Binary Counter



BCD Synchronous Counter

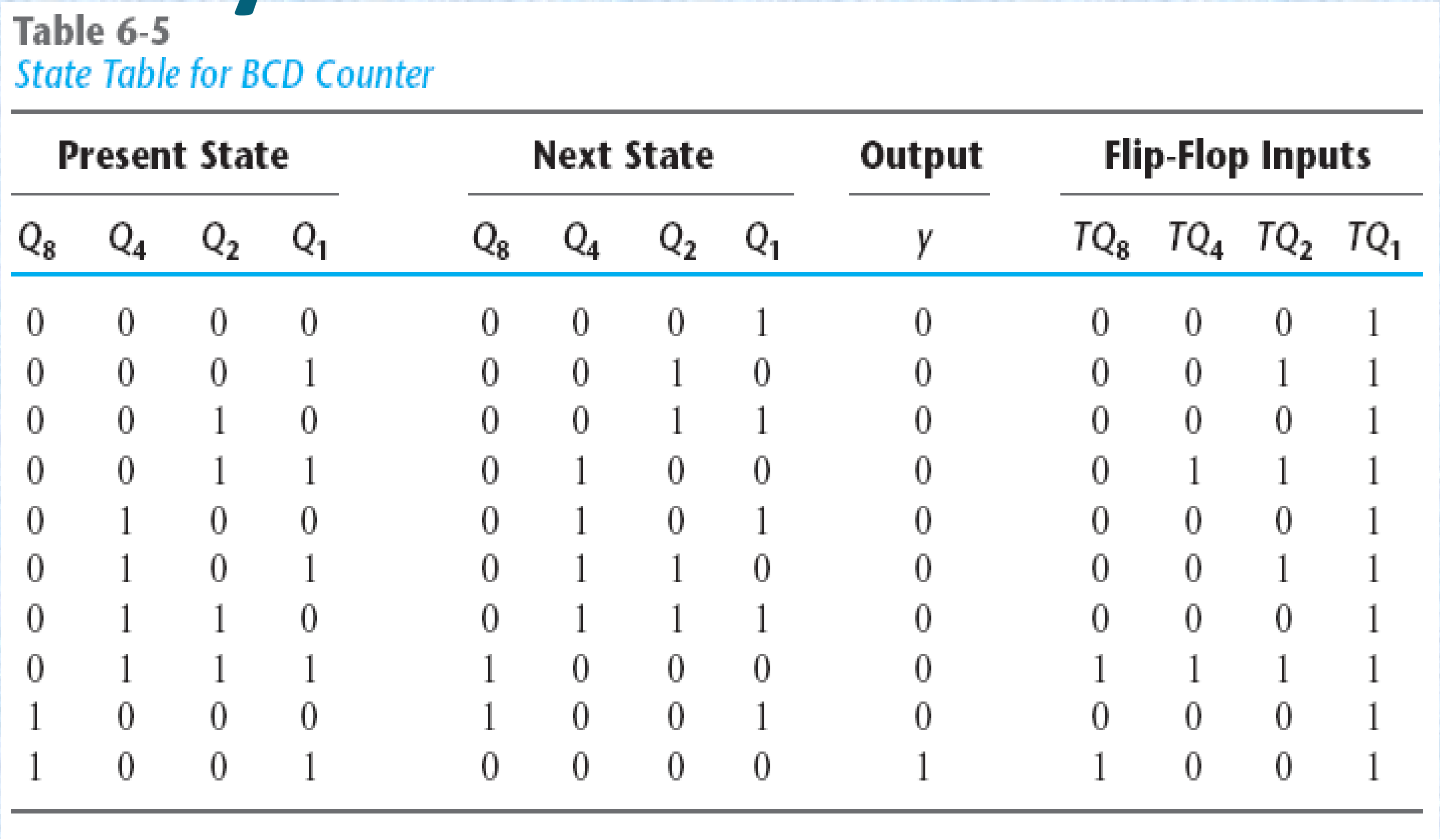
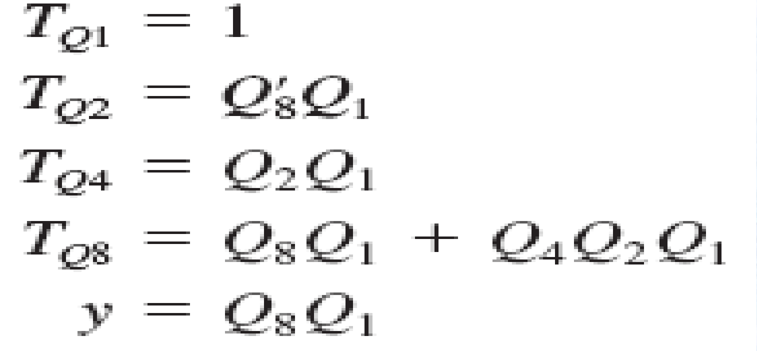


Table 6-5
State Table for BCD Counter

Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1



$$
\begin{aligned}
T_{Q1} &= 1 \\
T_{Q2} &= Q'_8 Q_1 \\
T_{Q4} &= Q_2 Q_1 \\
T_{Q8} &= Q_8 Q_1 + Q_4 Q_2 Q_1 \\
y &= Q_8 Q_1
\end{aligned}
$$

Binary Counter with Parallel Load

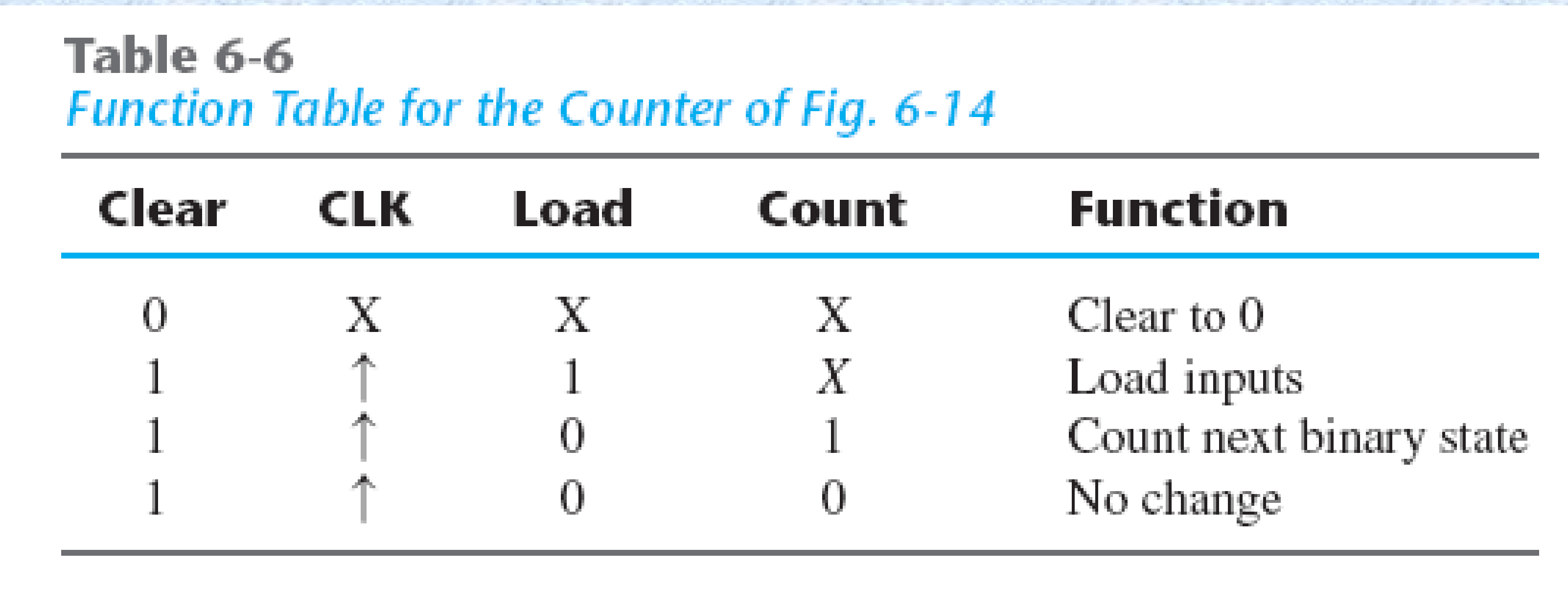
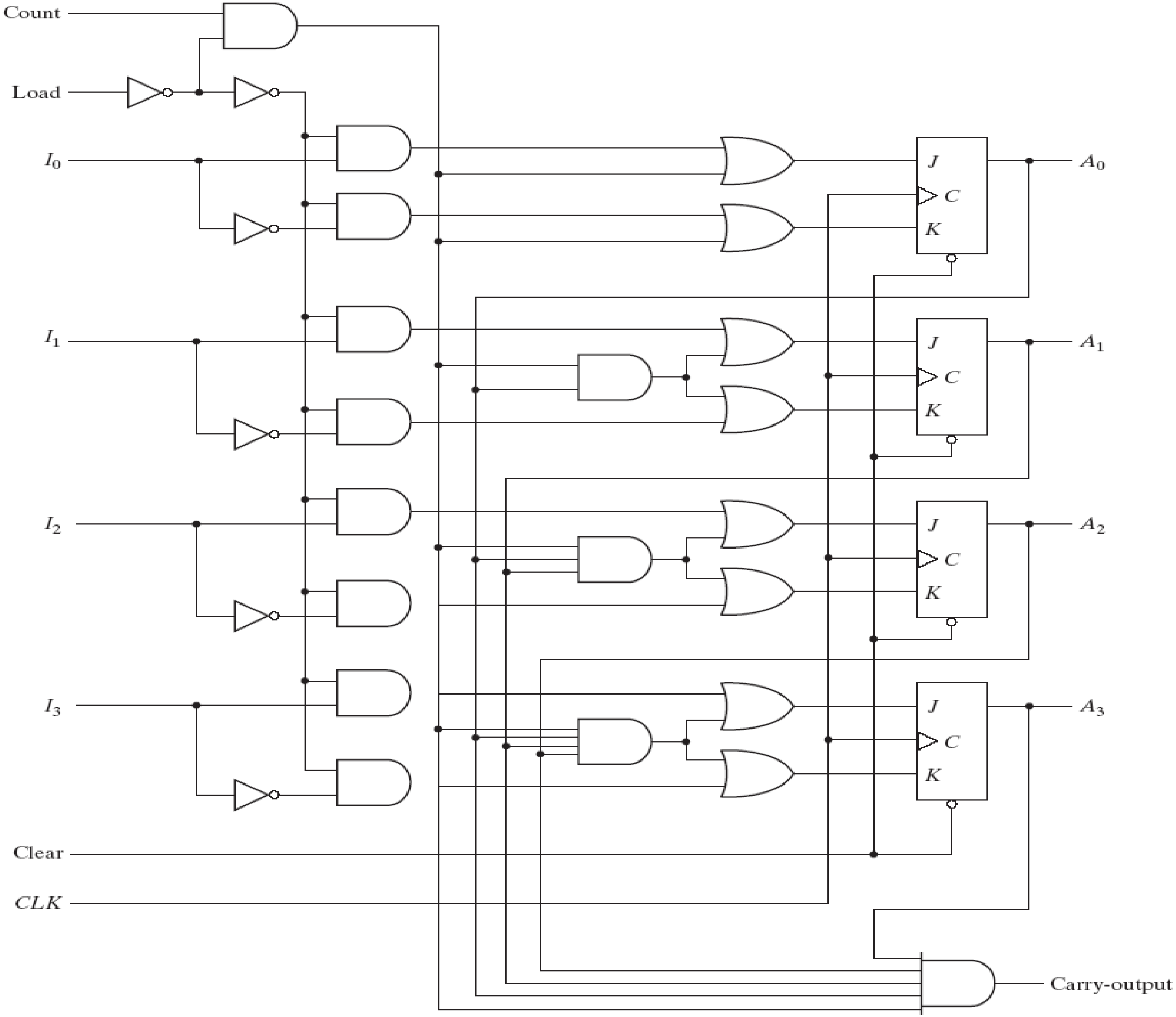


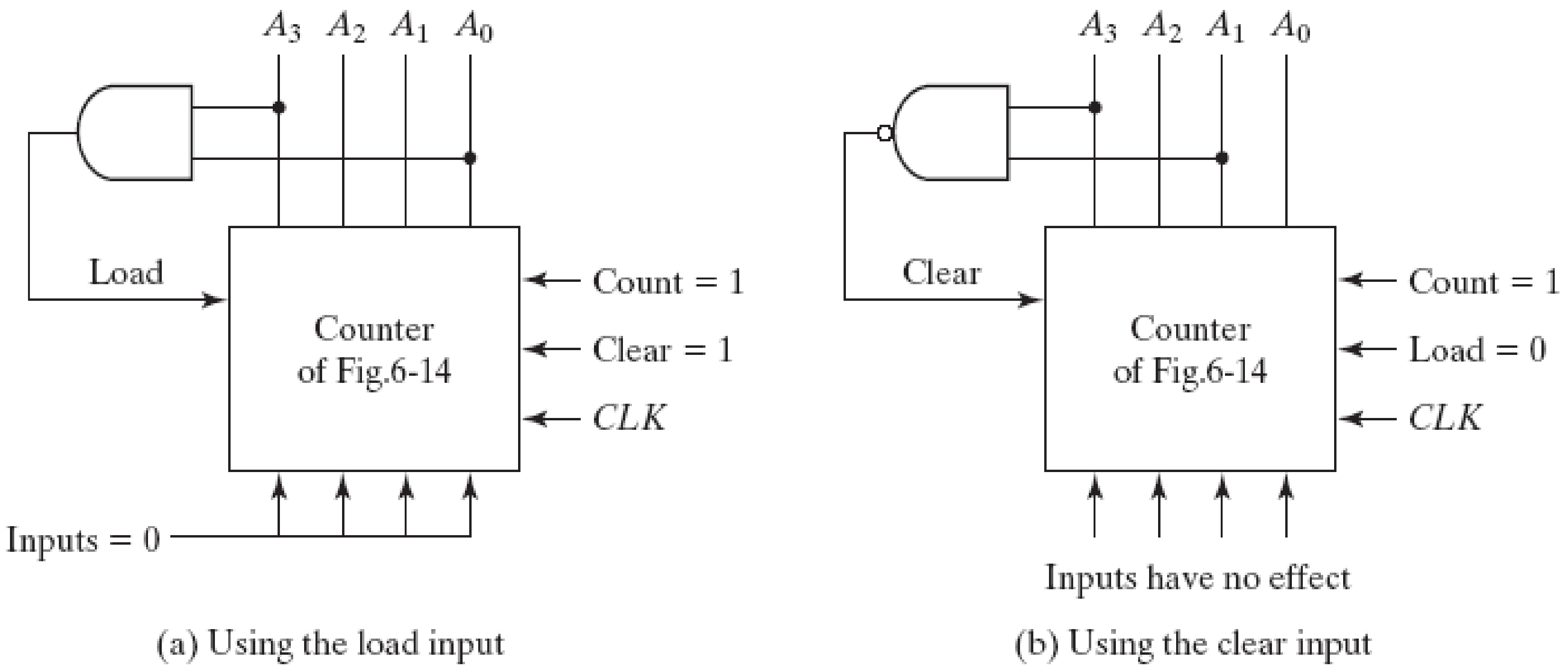
Table 6-6
Function Table for the Counter of Fig. 6-14

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	*X*	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change



Count
Load
I_0
I_1
I_2
I_3
Clear
CLK
J
C
K
J
C
K
J
C
K
J
C
K
A_0
A_1
A_2
A_3
Carry-output

FIGURE 6-14



(a) Using the load input

(b) Using the clear input

FIGURE 6-15

Two ways to Achieve a BCD Counter Using a Counter with Parallel Load

Counter with Unused States

- A circuit with *n* flip-flops has 2^n binary states.
- There are occasions when a sequential circuit uses less than this maximum possible number of states.
- States that are not used in specify-ing the sequential circuit are not listed in the state table.
- When simplifying the input equa-tions, the unused states may be treated as don’t-care conditions or may be assigned specific next states.
- Once the circuit is designed and constructed, outside interference (like radio waves) may cause the circuit to enter one of the unused states.
- In that case, it is necessary to ensure that the circuit goes back into one of the valid states so it can resume normal operation. Otherwise, if the sequential circuit circulates among unused states, there will be no way to bring it back to its intended

Example:

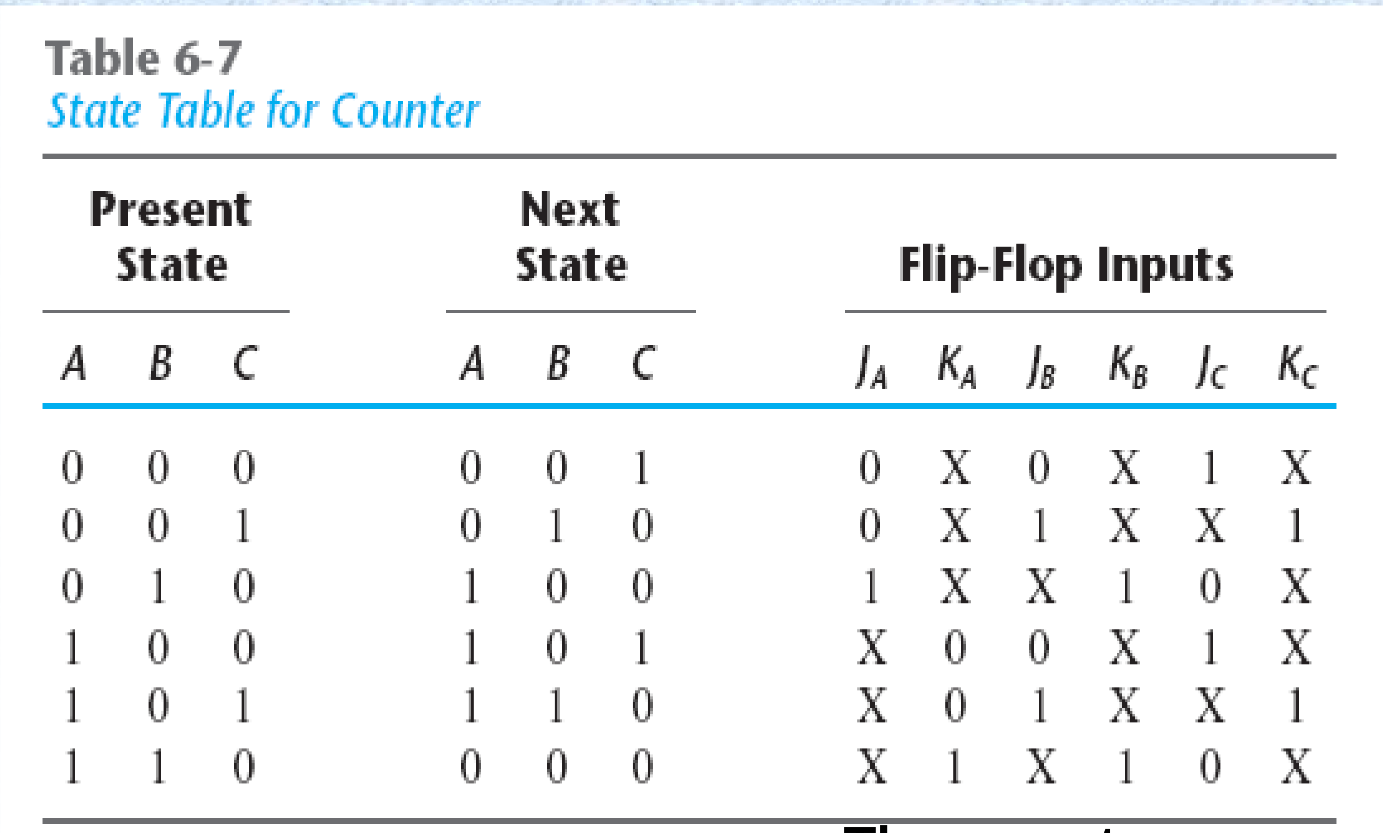
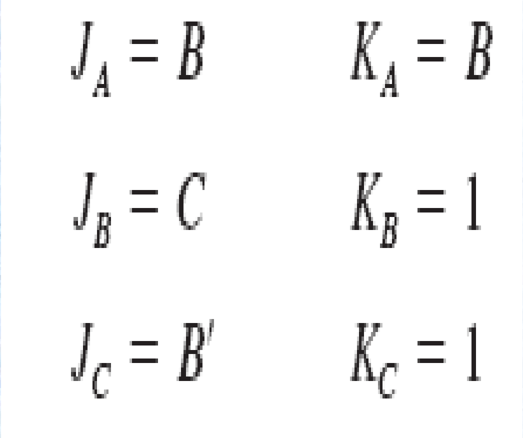


Table 6-7
State Table for Counter

Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

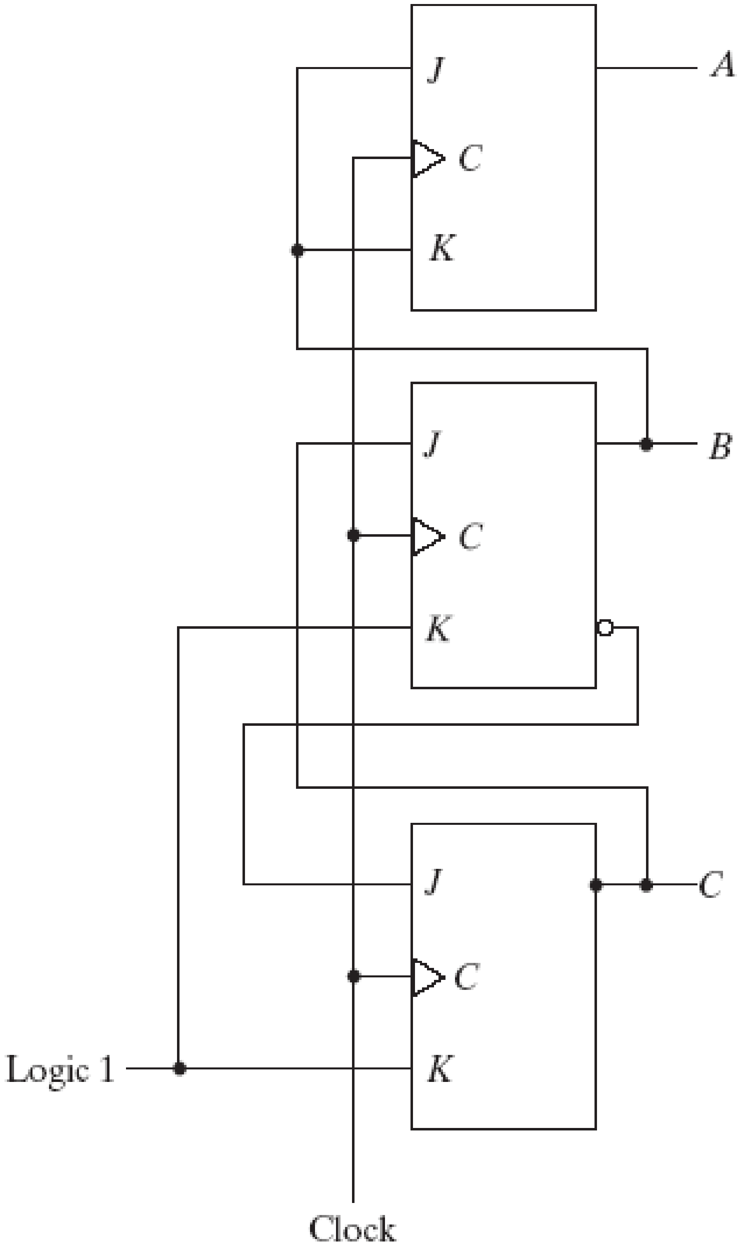


$J_A = B \qquad K_A = B$

$J_B = C \qquad K_B = 1$

$J_C = B' \qquad K_C = 1$

There are two unused states (011) and (111), we analyze the circuit to determine their effect.

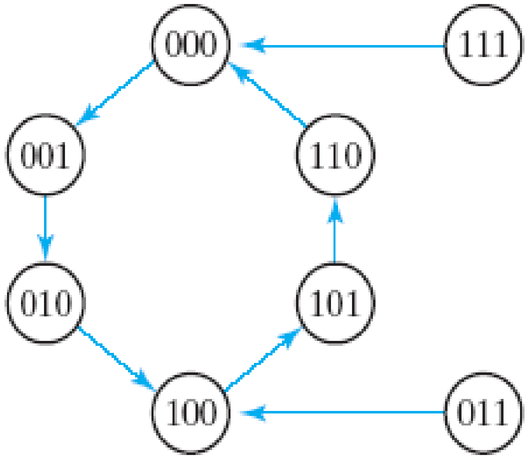


(a) Logic diagram

$J_A = B \qquad K_A = B$

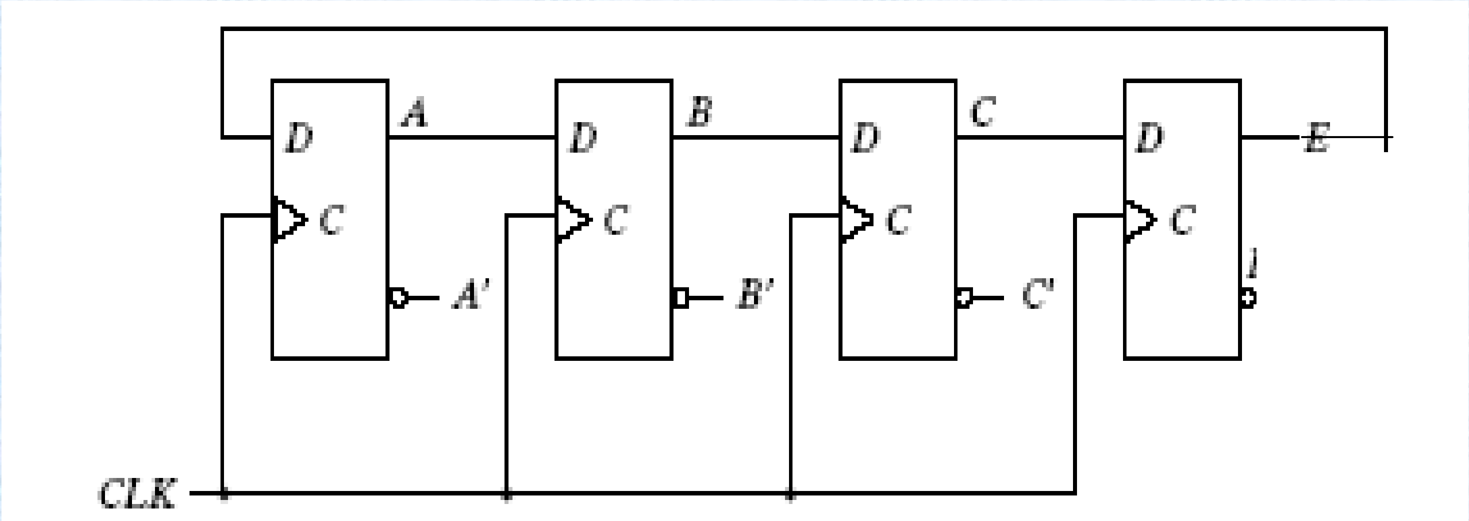
$J_B = C \qquad K_B = 1$

$J_C = B' \qquad K_C = 1$

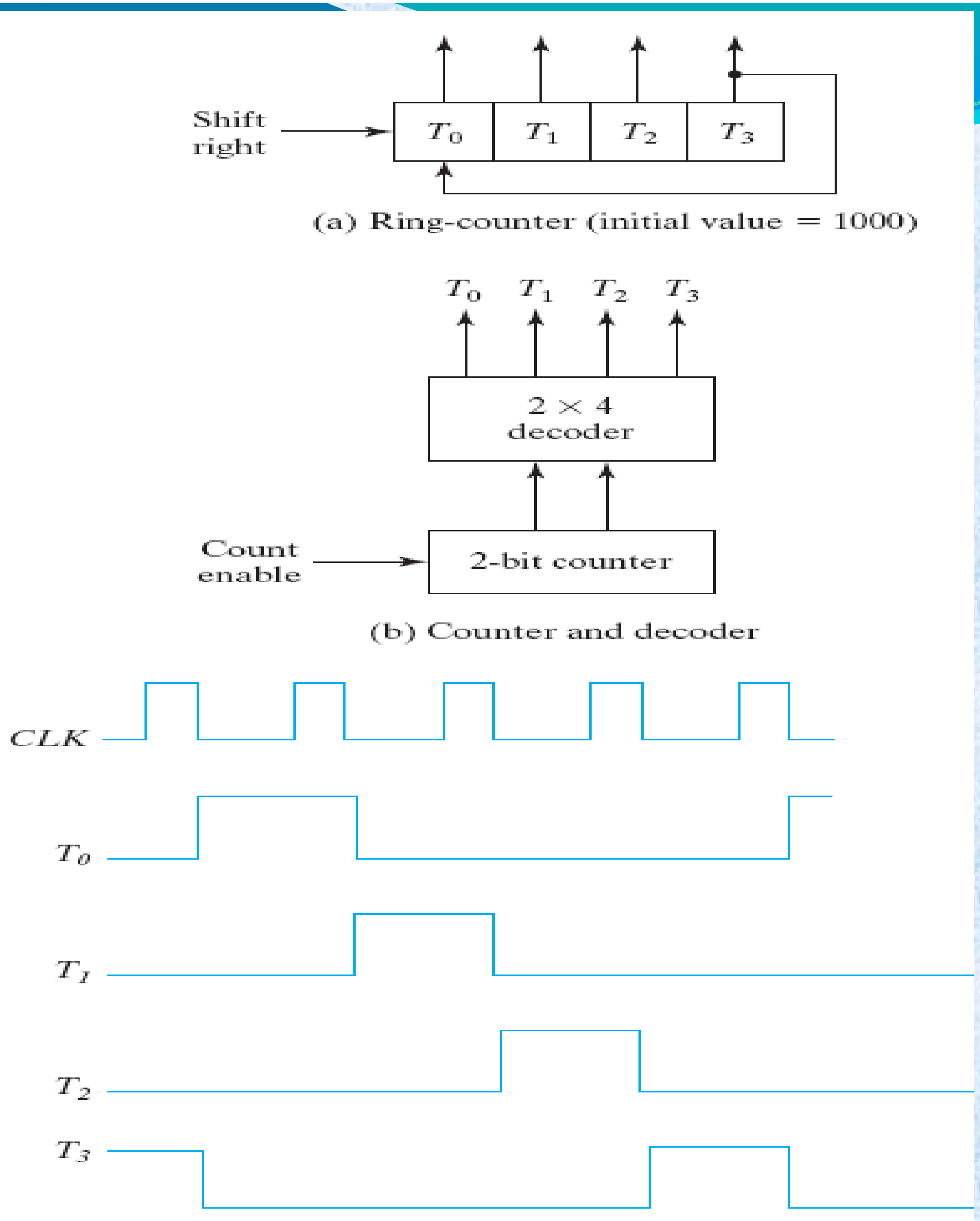


(b) State diagram

Ring Counter



Sequence number	Flip flop outputs A	 B	 C	 E
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

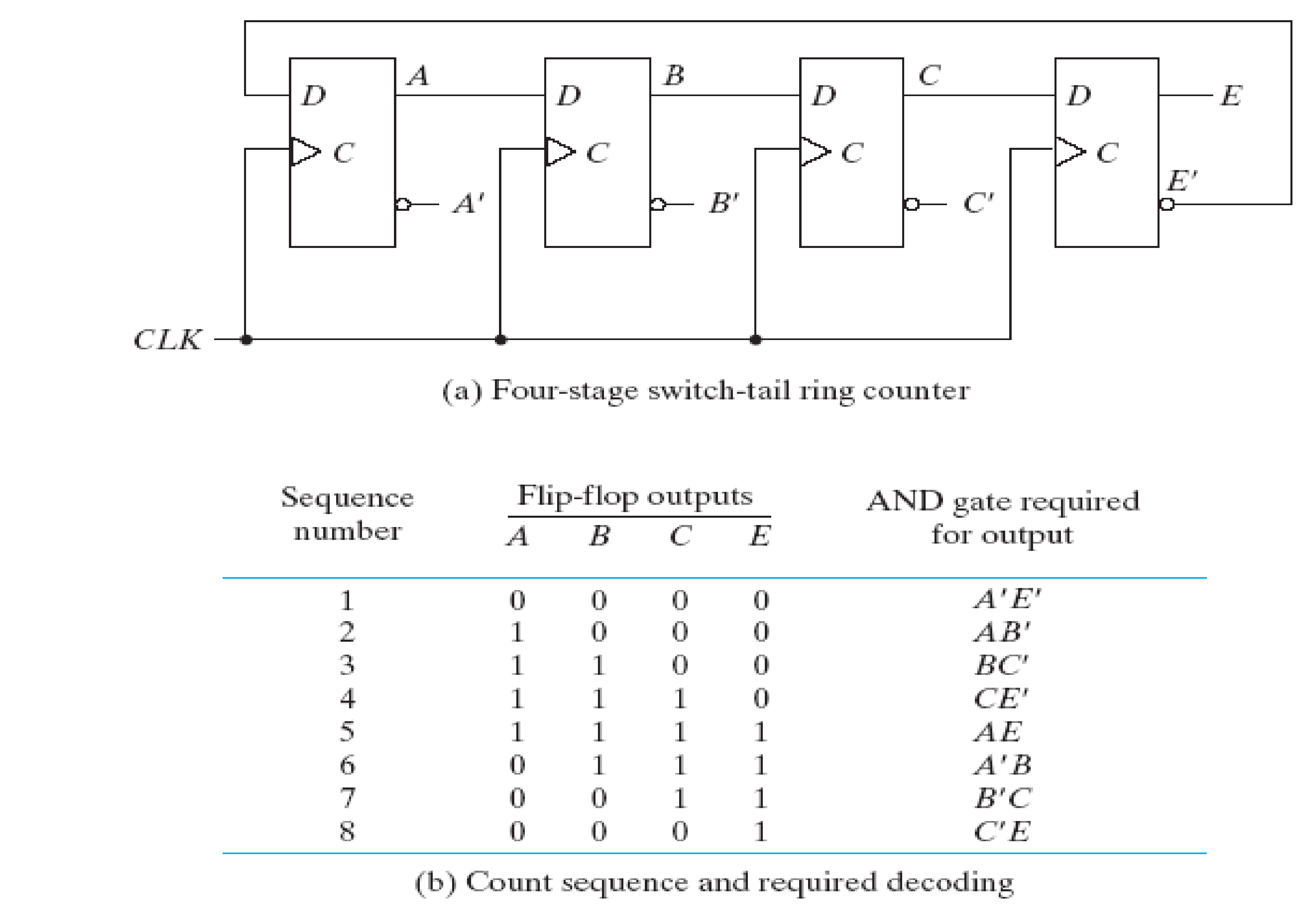


(a) Ring-counter (initial value = 1000)

(b) Counter and decoder

(c) Sequence of four timing signals

Johnson Counter



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs				AND gate required for output
	A	*B*	*C*	*E*	
1	0	0	0	0	*A′E′*
2	1	0	0	0	*AB′*
3	1	1	0	0	*BC′*
4	1	1	1	0	*CE′*
5	1	1	1	1	*AE*
6	0	1	1	1	*A′B*
7	0	0	1	1	*B′C*
8	0	0	0	1	*C′E*

(b) Count sequence and required decoding